CRT-2001-20

Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

DESIGNING DISTRIBUTION NETWORKS: FORMULATIONS AND SOLUTION HEURISTIC

by

Sophie D. Lapierre
Angel Ruiz
Patrick Soriano

May 2001

102-02940/

PUBLICATION
CRT-2001-20

DESIGNING DISTRIBUTION NETWORKS: FORMULATIONS AND SOLUTION HEURISTIC

by

Sophie Lapierre[1]
Angel Ruiz[2]
Patrick Soriano[3]

* This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

[1] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7, and Département de mathématiques et de génie industriel, École Polytechnique, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7 (sophiel@crt.umontreal.ca)

[2] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7, and UT Compiègne, B.P. 20529, 60205 Compiègne Cedex, France (angel@crt.umontreal.ca)

[3] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7, and Service de l'enseignement des méthodes quantitatives de gestion, École des HEC, 3000, chemin de la Côte-Ste-Catherine, Montréal, Canada H3T 2A7 (patrick@crt.umontreal.ca)

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2001

RÉSUMÉ

Le développement du transport ainsi que l'utilisation des centres de consolidation ont changé le portrait de la logistique dans les trois dernières décennies. Les centres de consolidation ont permis de mieux remplir les conteneurs en termes de poids et de volume. De plus, la possibilité de choisir entre plusieurs transporteurs, voire de combiner leurs services avec la gestion d'une flotte de véhicules privée, peut rapporter une réduction des coûts et une amélioration du niveau de service aux clients. Cependant, la complexité des décisions à prendre a fortement augmenté. En conséquence, les modèles existants doivent être améliorés afin de tenir compte des nouvelles opportunités. Nous révisons d'abord les différents modèles et formulations proposés dans la littérature pour les réseaux de distribution avec centres de consolidation. Ensuite, nous présentons un nouveau modèle qui détermine le nombre et la localisation des centres de consolidation à utiliser ainsi que la meilleure alternative de transport – LTL, FTL, courrier ou une flotte privée – pour chaque trajet à parcourir. Pour résoudre ce problème, nous proposons une heuristique hybride combinant la recherche avec tabou et la recherche à voisinage variable. La performance de l'algorithme est évaluée sur un ensemble de problèmes de différentes tailles et demandes aléatoires, mais utilisant des distances et structures de coûts réelles telles que publiées par un transporteur des États-Unis. Pour les problèmes de taille réduite, les résultats de l'heuristique sont comparés aux solutions optimales trouvées à l'aide d'une méthode exacte.

Mots-clés : réseaux de distribution, centre de consolidation, métaheuristiques, recherche tabou, recherche à voisinage variable.

ABSTRACT

The fast development of transport activities and the introduction of shipment consolidation have considerably changed the logistics context over the last three decades. Consolidation terminals, also called transhipment centers (TC) or hubs, have justified their presence by improving the loading of trucks in terms of both volume and weight. In addition, the possibility of using external carriers, exclusively or in coordination with a private fleet, can reduce costs and increase customer service. The right combination of these strategies can dramatically impact the cost of transport. However, the complexity of the decisions has also increased and existing models have to be improved in order to tackle these new challenges. In this paper, after discussing the different formulations for distribution networks with transhipment centers existing in the literature, we present a new model that determines the number and the locations of TCs as well as the best transportation alternative – LTL, FTL, Parcel, or own fleet – on each segment. To solve this problem, we present a very efficient heuristic combining tabu search and variable neighborhood search. The performance of this approach is evaluated on problems of various sizes generated with random demands but using real travel distances and cost structures (as published by a US carrier). For small sized instances, the heuristic solutions are compared to optimal ones obtained by an exact method.

Keywords : distribution networks, metaheuristics, tabu search, variable neighborhood search.

1 Introduction

For a long time, the field of logistics was limited to the day-to-day operations of transfering goods from source locations to customers without much consideration given to strategic issues. Things changed dramatically when operations research showed that well designed logistic systems could be instrumental in a company's success (Camm et al. 1997) which resulted in the increased interest for transportation research over the last three decades. However, the large number of available outside carriers and alternative transportation modes in present day transport operations, as well as the widespread use of transhipment centers (TC) have introduced new and difficult challenges when optimizing the design of distribution networks (Hall 1985). The aim of this paper is to address these important issues.

Geoffrion and Graves (1974) were among the first to investigate the use of intermediate distribution facilities between production plants and customers. They showed that the use of these facilities allows the concentration of shipments into a reduced number of "inter-hub" routes, increasing the flow on these links and consequently generating economies of scale by improving the load of the vehicles up to their capacities. Consider the case shown in Figure 1 where six shipments w_{ij} are sent from two suppliers i and three customers j. An alternative to shipping directly to each customers would be to

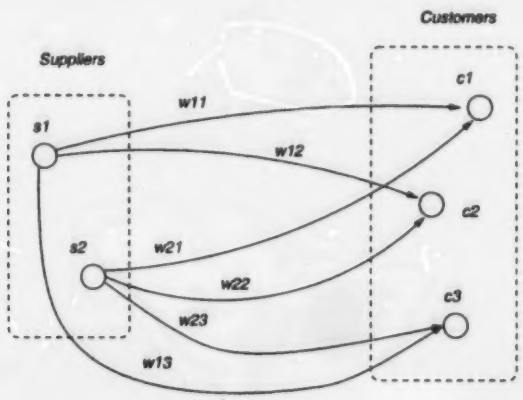

Figure 1: Original distribution network

ship sometimes via TC k as shown in Figure 2. This way, supplier s_1 and s_2 consolidate shipments for customers c_1 and c_2 through transhipment center $h1$ but ships directly to customer c_3. Then, at TC $h1$, shipments are split and reconsolidated according to their final destinations. The use of a TC increases traveling distance and manutention

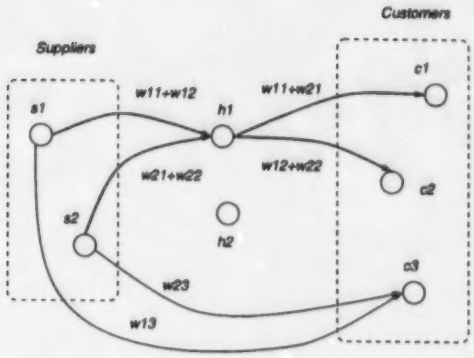

Figure 2: Distribution Network

for each shipment — there is no saving at the individual package level — but better truck loading can lead to overall savings. The geographic location of the common customers, the potential location of TCs, the transportation cost structure, and the heterogenity of shipments in term of volume and weight are the most important factors affecting the design of a distribution network. We elaborate on the latest two factors.

For transportation, distribution managers deal with numerous alternatives at their disposal for shipping decisions. They often can resort to their own fleets and/or a variety of alternate options through subcontracting to commercial carriers. In this contex, special attention must be given to the selection of the most appropiate option for each route. Commercial carriers deal either into FTL (Full-TruckLoad), LTL (Less-than-TruckLoad), or parcels. Each of these modes uses different types of vehicules and posts different rates depending on the transport segment to be travelled and the quantity of the goods to be shipped. Figures 3.a to 3.c show respectively typical rate tables from parcel, LTL and FTL carriers for a given class of goods. When comparing all the different transport options on a given segment for a particular shipment, it is therefore relatively easy to determine the most economical: the parcel carrier for small weight, FTL for large shipments, and LTL for intermediate weight. A compound cost function merging the best use of each category of carrier is illustrated in figure 3.d. Hall (1985) suggests to use such cost function on each route to design distribution network since it enables the integration of both routing and carrier/mode selection into a single model. However, this general cost function structure is non convex and non concave. The complexity of the problem is such that exact solution approaches are totally impractical for realistic sized instances (Klinzewicz, 1990). Little work has been

done to develop efficient heuristics to solve this problem since the research community has focused on special cases of this model (Campbell, 1994).

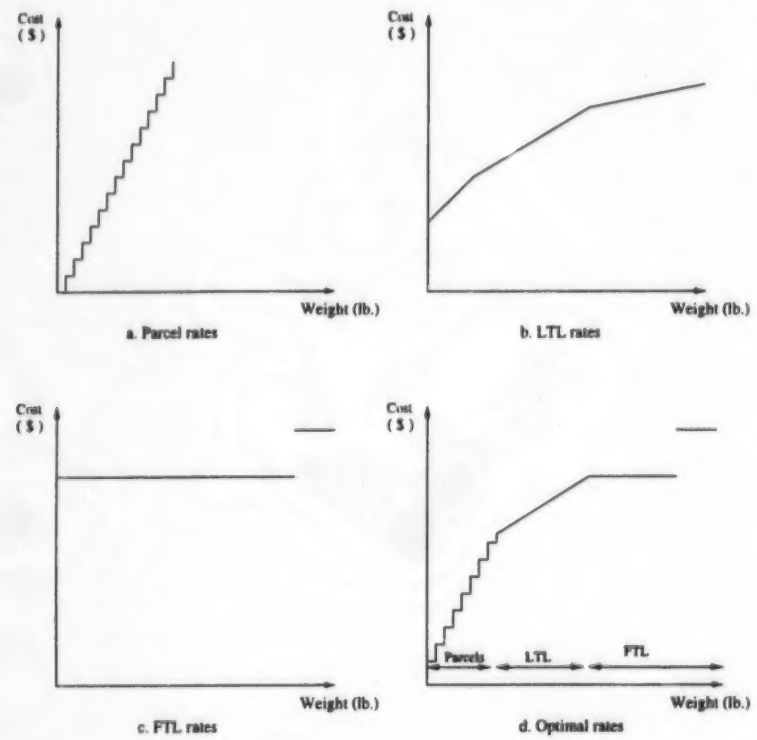

Figure 3: Function cost for Parcel, LTL and FTL carriers for a given class of goods

From another standpoint, the simultaneous consideration of shipment attributes (i.e., weight and volume) may lead to important savings when heterogeneous packages can be transported together (Hall 1991). In the FTL context, where containers are defined by weight and volume capacities, the right combination of packages can reduce the number of trucks needed by filling them to their weight and volume limits. In the case of LTL carriers, a price schedule based on weight is available for each product class — according to the National Motor Freight Traffic Association guidelines in North America — that is mainly determined by its weight density. Finally, parcel companies have a unique price schedule based on weight but they apply a penalty called "dimensional weigth" when the package density is under a given threshold value. Therefore, as will be shown in section 3, the compound cost function of Figure 3.d should be extended to integrate

the volume attributes of shipments.

The aim of this paper is to present a general model that can serve from the perspective of both the distribution manager and the carrier operator. This model accounts for the above mentioned particularities of modern complex transportation networks : (1) the potential use of transhipment centers and (2) the use of general transport cost functions that enable integration of carrier/mode selection and the consideration of shipment attributes of weight and volume. In this context, the distribution network design problem consists in determining the optimal number and location of facilities (TCs) as well as the choice of carriers on each route while deciding at the same time which shipments must be consolidated at each TC according to the weight and volume of the total freight.

To solve this difficult problem, we propose a heuristic procedure based on a combination of tabu search and variable neighborhood search methods. We then evaluate the performance of this procedure on a wide set of realistic instances by comparing the solution it provides to optimal ones or lower bounds obtained by an exact approach. As we will see, our algorithm provides good solutions to small test problems and is able to solve large-scale problems.

The remainder of this paper is organized as follows. Section 2 presents a review of the related literature. Then, a mathematical model is proposed. Some background on tabu search and variable neighborhood search is provided in section 4 which then describes the proposed heuristic. Computational results are presented in section 5. Finally, some avenues of future research are discussed in section 6 which concludes the paper.

2 Overview of previous related work

Distribution network design problems have received an increasing attention from the research community in recent years since great savings are expected from better designed network (e.g., Camm et al. 1997). Work has been performed simultaneoulsy at the modeling and solving level.

Early work on designing distribution network focused on locating warehouses in respect to customers. The warehouse location problem is the ancestor of the distribution network design since it accounts for costs associated the the use of a warehouse and the transportation costs from the warehouse to the customers but it does not account for the transportation costs between the suppliers and the warehouses (for more details on the warehouse location problem, see Love et al. 1988). Accounting for the location of

suppliers increases the complexity of the problem and brings it to the class of problems of "network design".

Three approaches have been used to model the distribution network design problem. The first one uses general cost functions to account for return to scale on a given route. Burns *et* al. (1985) study networks with a structure similar to the one considered here for the simpler single hub location case. In this category, the model presented by Klincewicz (1990) is the most detailed one, using general concave transportation cost function and a manutention cost function when using a TC center. However, only a simplified version of this model is solved since the basic model is too complex to solve.

A second approach to design distribution network uses only linear transportation cost function by using rates for each routes according to historical costs or a projected use of parcel, LTL and FTL carriers. This approach has the advantage to be easily solved with a general *Minimum cost flow* algorithm or a generic optimizer software and is therefore the typical approach proposed today by commercial supply chain software such as the CAPS Logistics Supply Chain Designer and Ilog Supply Chain Management Software. In this category, specific work has been performed to locate TCs when a discount rate is given for freight transported between TCs (referred as hubs). This approach is appropriate to design the network of a LTL carrier. For work into that direction, see O'Kelly (1992), Campbell (1994), Aykin (1994), Ernst and Krishnamoorthy (1996, 1998), Skorin-Kapov *et* al. (1997) and Pirkul and Schilling (1998), Ebery *et* al. (2000), O'Kelly*et* al. (1996), and O'Kelly and Bryan (1998).

Finally, the last strategy is to solve the freight transportation problem. This approach is quite good to explicitily considering both shipments weight and volume. Hall (1991) studies the case where commodities are characterized by two attributes (weight and volume) and where the objective is to obtain a good mix of heavy and light goods in order to improve the utilisation of the vehicles capacity. This model is well suited for FTL applications in which customers pay for a whole container even when it is not loaded to its full capacity. A good realistic analysis of the freight transport problem is presented by Fleischmann (1993). Barnhart et al. (1993) model a distribution network where containers can be filled to their weight (weight-out) or volume (cube-out) capacity as a MIP. Several (i.e. three) transport modes are available between transhipment centers and customers. This approach is accurate but require very good skills in modeling the problem in order to solve it with commercial optimizer software.

The modeling approach using general transportation cost function on each route seems the most promising strategy. This approach is global and can fulfill the various need to design a distribution network. However, we must find a way to solve such problem.

The classic strategy to solve such a problem is to linearize the cost function by adding variables. Zangwill (1968) proposes some theorems for solving the minimun cost flow in certain concave cost networks. Hall (1989), building on previous work by Yaged (1972), defines some optimality criteria and introduces the notation of *shared* and *exclusive* arcs in order to classify respectively arcs where a transfer of flow from other routes reduces the global per unit cost of the network and those where that cost is increased. Unfortunatly, the application of these criteria is limited to the case where cost functions are continous and differentiable. Vogt and Even (1972) model concave cost functions from real railroad and truck tariffs as piece-wise linear functions and then solve the problem by a separable-convex program. However, this approach generates a number of variables so large that it becomes computational intractable even for medium sized problems. Non-convex cost functions have also been studied by Barton and Rech (1970) in a transportation context with several modes of transportation and transhipment nodes. In this approach, a *linear-convex* approximation is performed on the non-convex function. Then branch-and-bound (B&B) followed by a linear approximation are sequentially applied until both the real and the linearized rate for the solution flow are sufficiently close. Kelly and Khumawala (1982) use local linear approximations of the concave costs and then they solve iteratively transportation problems converging to a final solution. Powell and Sheffi (1989) present such a strategy for solving a LTL loading problem and Lamar (1993) for minimizing concave cost network flow problems.

But most paper present heuristics to solve the design problem with TC. Klincewicz (1990) proposes some basic heuristics that solve a simplified version of the freight transportation problem in which only one portion of the network (either the origin–to–hub or hub–to–customer side) presents concave cost functions while the other is considered linear. Klincewicz (1991) decomposes the problem into master (hub location) and slave (hub allocation) subproblems and uses simple "exchange" heuristics in order to solve them iteratively. Then Klincewicz (1992) and Skorin-Kapov and Skorin-Kapov (1994) both proposed approaches in which tabu search is applied to each subproblem iteratively. More work needs to be performed to develop metaheuristics that can well solve such complex network problem.

3 Mathematical formulation

We propose a distribution network design model that differs from previous ones on at least one of the following aspects since it (1) proposes a general formulation that integrates several transportation modes –FTL, LTL, Parcel and private fleet–; (2) accounts for real cost functions, considering both weight and volume of the shipments;

(3) allows direct shipments between suppliers and customers and multiple allocation for non-hub nodes; (4) considers the number of open hubs as an endogeneous decision variable whose value is determined when solving the problem.

Let us consider a general freight distribution network where commodities have to be transported from several source nodes to different destinations. Each origin-destination pair is considered as a different commodity (i.e., a demand) and each commodity must use a single route (i.e. single routing for each demand). Commodities can travel either directly from their origin to their final destination node or, alternatively, they can be routed through one of the available hubs. In the latter case, individual commodities having the same origin can be grouped into a "shipment" and sent together towards a given consolidation terminal. There, arriving shipments are "broken down" and individual commodities can be recombined with others coming from different origins but sharing the same final destination thus forming new shipments for the second part of their travel.

Figure 3 showed how a compound function can model transportation costs when several modes or carriers are available. As it was illustrated, the general cost function in 3.d is essentially dependent of the LTL cost structure. We will therefore use the cost structure coming from this context in the rest of this study, the introduction of any additional mode or carrier affecting only the values of the parameters in a portion of the cost function. In the case of LTL carriers, the National Motor Freight Traffic Association in conjunction with the National Classification Committe, publishes the National Motor Freight Classification (NMFC) that categorizes shipments according to both their density and commercial value. This classification is used by LTL carriers in order to generate rate tables for each of these categories. For example, Table 1 shows LTL rates for two NMFC classes (class 55 and 70 for products having densities higer that 35 and 15 pounds per cubic feet respectively) for a given route as published by a commercial American LTL carrier.

Rates from ZIP 10001 to ZIP 94001 ($ per 100 pounds)									
Class	<500 pounds	>500 pounds	>1000 pounds	>2000 pounds	>5000 pounds	>10000 pounds	>20000 pounds	>30000 pounds	>40000 pounds
70	153.82	124.6	106.1	89.24	76.95	63.05	55.37	47.67	44.64
55	129.57	104.9	89.43	75.17	64.82	53.13	46.65	40.15	37.58

Table 1: LTL rates from New York to Los Angeles

Rate-schedule such as Table 1 are divided into columns (weight range bounded by fixed "break-weights") and lines (classes according to the NMFC). The rate for a given product is therefore obtained from the rate-schedule by simultaneously selecting the class and

the weight range corresponding to the product's class and weight. Costs generated from Table 1 present some discontinuities on the weight break-points. The rate calculator must therefore compare the resulting cost with the rate from the subsequent weight category multiply by the lowest weight from that category and then take the lowest price. The typical LTL cost functions look like the one shown in Figure 4. Although price schedule of LTL carriers are given according to a table, we will assume that it is always possible to retrieve the underlying cost function shown in Figure 4. Klincewicz (1990) uses also this underlying cost function in his model to solve its freight transportation problem but limited his case to the unique product class and did not allow "plateau" in the cost structure.

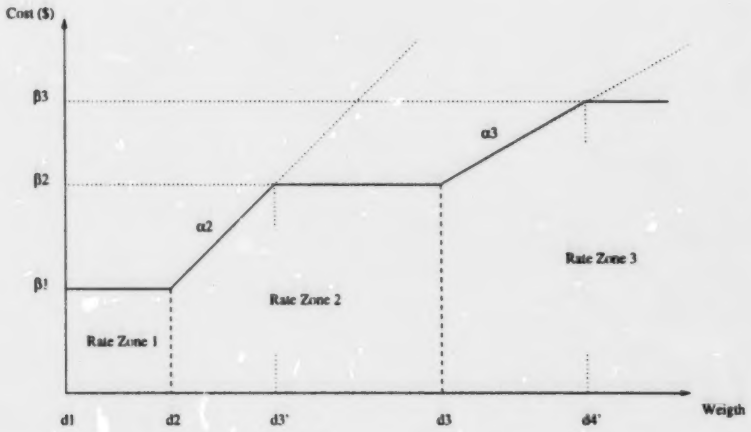

Figure 4: Applied cost to weight function for a given travel and a given class

Let us consider w as the weight of a given commodity belonging to class c and rate-zone z according to its weight. Therefore, the pair (c, z) determines the rate applicable to this commodity which is defined by α and β, respectively the slope and the intercept with the "y" axis of the corresponding portion of the cost function. The transportation cost for the shipment of an individual commodity is then computed as $c = \alpha w + \beta$. If several items travel together as a group on a given network segment, let W^c be the summation of the weights of all the items in the shipment belonging to class c, and Ω the total weight of the shipment. The total transportation cost becomes $C = \sum_c (W^c \alpha^c + \beta^c)$, where α^c and β^c are based on the rate-zone Z corresponding to Ω, the total weight of the shipment travelling on the segment.

Let i, j, and h, respectively be the indices for the origin I, destination J and hub H node sets and let us also define the following decision variables :

x_{i0j} $\begin{cases} 1 & \text{if commodity from origin node } i \text{ to destination node } j \text{ travels directly;} \\ 0 & \text{otherwise;} \end{cases}$

x_{ihj} $\begin{cases} 1 & \text{if commodity from origin node } i \text{ to destination node } j \text{ travels through hub } h; \\ 0 & \text{otherwise;} \end{cases}$

y_{ij}^c $\begin{cases} 1 & \text{if any commodity of class } c \text{ travels from any node } i \text{ to any node } j; \\ 0 & \text{otherwise;} \end{cases}$

The model can then be written as follows :

$$\text{Minimize} \quad \sum_I \sum_J x_{i0j}(w_{ij}\alpha_{ij} + \beta_{ij}) + \sum_I \sum_H \sum_C (W_{ih}^c \alpha_{ih}^c + y_{ih}^c \beta_{ij}^c) + \sum_H \sum_J \sum_C (W_{hj}^c \alpha_{hj}^c + y_{hj}^c \beta_{hj}^c) \tag{1}$$

$$\text{Subject to} \quad x_{i0j} + \sum_H x_{ihj} = 1 \quad \text{for all } i, j \tag{2}$$

$$W_{ih}^c = \sum_J x_{ihj} w_{ij}^c \quad \text{for all } i, h, c \tag{3}$$

$$W_{hj}^c = \sum_I x_{ihj} w_{ij}^c \quad \text{for all } h, j, c \tag{4}$$

$$y_{ih}^c M \geq W_{ih}^c \quad \text{for all } i, h, c \tag{5}$$

$$y_{hj}^c M \geq W_{hj}^c \quad \text{for all } h, j, c \tag{6}$$

$$w_{ij}, W_{ij}^c \geq 0 \quad \text{for all } i, j, c \tag{7}$$

where M is a constant of arbitrarily high value. The objective function (2) comprises three summation terms : the global cost for the direct shipments, the cost of shipments from origins to hubs and the cost of shipments from hubs to destinations. To simplify the formulation, the class indices for the α and β parameters inside the direct shipments term have been dropped considering that shipments of an individual commodity involve only the class associated to the given commodity. For the consolidated terms, α^c and β^c are obtained from rate-tables for each class using the rate-zone given by the total weight on the segment, $\Omega_{ij} = \sum_C W_{ij}^c$. Constraints (3) ensures that each demand will be assigned to an itinerary. Constraint (4) and (5) determine the consolidated weight, for each class, travelling from source nodes to hubs in the first case and between hubs and destination nodes in the later. Constraints (6) and (7) are needed because of the flat segments in the cost functions.

Introducing the rate-zone selection into the mathematical program is a simple task that will not be described in this paper. However, it is important to recall that the number

of integer variables required in order to represent such a piece-wise linear function is equal to the number of linear "pieces" in the function. The number of integer variables in the model increases very quickly with the size of the problem therefore limiting in practice the size of the problems that can be solved optimally.

This approach can also be useful for LTL carriers in order to optimize their operations and, in consequence, reduce their costs. Carriers calculate the rates showed in their tables based on a given transportation policy that most of the times assumes a single transshipment in the route. If provided with real transportation costs, the solution to the model presented in this section can therefore be used to evaluate if the present tarifs and distribution decisions are indeed "optimal" and, if not, to guide the carrier on how to adjust them.

4 The Tabu Search with Variable Neighborhood Metaheuristic

The hybrid metaheuristic merges a Variable Neighborhood Search (VNS) approach with a Tabu Search (TS) algorithm. TS explores a feasible region of solutions using a descent strategy guided by several control rules. A detailled description of the basic TS Heuristic can be found in Glover (1989 and 1990). On the other hand, Variable Neighborhood Search is a new approach that has provided very promising results in solving large combinatorial optimization problems (Hansen and Mladenović, 1998 and 2000). Synergies can be obtained from the fact that TS could be seen as an improved local search tool in a VNS scheme and, reciprocally, VNS can be used as a diversification-intensification tool within a TS framework. We will briefly describe the main components of the TS structure before discussing in more detail VNS definitions and the implementation of the main search algorithm.

4.1 TS Components

The main components of the TS algorithm are a neighborhood definition, search memories or rules (tabu list, frequency list and aspiration criteria), a diversification procedure and a stopping criterion. We will discuss neighborhood definition in section 4.3.

In a TS framework, search is guided by both short term and long term memories. The short term memory, called tabu list, is used to avoid cycling. It records the most recent moves in order to forbid their reversal. For example, when a "move" (i.e. a change on a part of the solution) is performed, the opposite move is declared

tabu for a predefined or pseudo-random number of iterations (*length_tabu*) to ensure that the move will not be reversed for some time. As tabu-variables are locked, we ensure that during this time the move will not be reversed. However, to prevent these restrictions from being too stringent, if a change on a tabu variable leads to a solution better than the best found so far, the change is allowed (aspiration criteria).

Long term memory, based on a *frequency list*, aims at dynamically diversifying the search. It encourages moves that have not yet been implemented or those that have been implemented with less frequency.

The TS heuristic calls a diversification after running local search for a given number of iterations (*it_div*) without improvement of the best recorded solution. When that criterion is met, it is assumed that the exploration of a region of the solution space has been completed and search is driven towards a new region. The new starting solution needs to be different enough from the solutions explored so far to guarantee that the exploration will really move from the present region but, at the same time, it is not desired to move towards regions with low potential. Search is finished after completing a given number of iterations (*it_total*) or search time (*time_total*), whichever is reached first.

The TS framework is applied to the problem studied here in a straightforward manner. A move is defined simply as the re-routing of a commodity. Three attributes (tabu, frequency and last_moved) are associated to each commodity. The tabu condition and the frequency list are managed using these attributes; when a move is performed, the tabu attribute of the affected commodity is set to the iteration number in which the tabu condition will expire (current_iteration + length_tabu). Each time that a commodity is re-routed, its frequency attribute is increased by one; the frequency attribute then acts as a penalty affecting the objective value during the neighborhood exploration. The current iteration number is also recorded in the attribute last_moved for a later use during the diversification procedure. Divirsification first selects a given number of commodities (size_div) having the lower frequencies. The search will restart from a new initial solution generated from the current solution by forcing random changes on the routing for the selected commodities. This strategy is expected to force changes in those parts of the solution that have remained unchanged for a long time (the skeleton of the current solution). Values for all the above mentionned parameters are set during a tuning-phase prior to the formal computational testing.

4.2 Variable Neighborhood Search

VNS proposes a systematic change of neighborhood during a local search. In this definition, "systematic" is the key word because VNS focuses on the topology of the

solution space rather than on the ability of a given descent heuristic to find an optimun. Expecting that a local optimum obtained in a topology defined by a given neighborhood will not necessarly be one in the topology defined by a different neighborhood, VNS would then avoid being trapped by local optima.

VNS is mainly composed of a local search tool, a set of different neighborhoods, a set of rules defining the use and the selection of neighborhoods according to the search status, and a stopping criterion. The local search tool is generally a simple descent heuristic, but it is also possible to imagine a VNS that could change systematically between a set of different heuristics and, therefore, different neighborhoods. The development of different neighborhood structures is a very important element of a VNS implementation. As different neighborhoods explore the space of solutions differently, we can expect that VNS would perform better when the available neighborhoods show different, rather than similar, structures. The definition of neighborhoods is problem dependent, but in practice, neighborhoods can be derived in a rather straighforward manner from a basic or natural neighborhood. Section 4.3 describes how k different neighborhoods can be easily defined for the problem studied here.

Rules for neighborhood selection and exchange are a key element in the VNS aproach. Different neighborhoods show different abilities and require different exploration times, and the main idea is to combine them in a way that simultaneously maximizes their synergies and minimizes the total computation cost. For exemple, two neighborhoods can form a intensification-diversification framework if the first neighborhood explores solutions very similars to the incumbent one (intensification) while the second, "larger", neighborhood allows much more different solutions to be explored (diversification). As larger neighborhoods cover, in general, a higher number of solutions, they are often very expensive in terms of computational effort. Therefore, the intesification-diversification framework would selected the larger neighborhood only when the search is supossed to be locked around a local optima, and only for the time required to ensure that the search process escapes from it. Several variations of the general VNS framework have been proposed by Hansen and Mladenović (1998, 2000). Section 4.4 presents the rules used by the heuristic in order to determine *when* a change of neighborhood is performed and *which* neighborhood to select.

4.3 Implementing neighborhoods

Let us define the "order" of a neighborhood as the number of allowed differences between two neighbor solutions. For exemple, the neighborhood of "order 1" for the problem under study here is defined as the set of solutions obtained by changing the routing path for one and only one of the commodities at a time. In other words, every two

neighbor solutions differ in the routing of one commodity. Larger neighborhoods are easily derived according to the order definition; the neighborhood of order 2 is built by considering solutions that differ simultaneously in the routing of 2 commodities and, in general, the neighborhood of order k includes solutions differing in the routing of k commodities.

The practical implementation of neighborhoods for solving the problem studied in this paper differs slightly from the previous definition. The reason is that the number of solutions included in a neighborhood for a given k, even when k=1, is very large. In consequence, the explicit evaluation of the objective function for the whole neighborhood is very expensive in terms of computational effort. Therefore the heuristic uses a pseudo-random sampling technique in order to reduce the number of solutions to evaluate. Instead of considering the complete set of commodities, the sampling strategy selects a given proportion of origin nodes. The subset of the neighborhood to be evaluated consists of all the commodities emanating from the selected origin nodes. This sampling procedure, that can also be seen as a partition of the problem, was shown to be more efficient than the random sampling of individual commodities. The strategy has been extended for neighborhoods of order k in the following manner : first, a subset of source nodes N is selected randomly. Then for every origin node $r \in N$, a second random choice t is made in the set of hubs currently used by r, including the dummy hub "0" for the direct shipments. For this pair (r,t), every possible combination of k commodities travelling from source node r to the selected hub t is re-routed through the rest of available alternatives. In other words, if S_{rt} is the set of commodities originating from node r and consolidated through hub t, the neighborhood of the current solution for given r and t is generated by routing sequentially through the rest of the allowed paths every element in the set of combinations of the elements of S_{rt} taken k by k. The following exemple illustrates the procedure. Consider a network having I=2 origins, J=3 destinations and K=2 potential transhipment centers. Consider also a feasible solution $S_0 = \{s_{11}, s_{12}, s_{13}, s_{21}, s_{22}, s_{23}\} = (0,0,0,1,0,2)$ where $s_{ij} \in K$ is the routing path for the commodity generated at node i and required by node j. Let $k = 2, r = 1$ and $t = 0$ be respectively the order of the neighborhood and the random selections for the origin node and the routing path. Then, the subset of commodities leaving from the selected source node r=1 and routed trough the consolidation center $t = 0$ is $\{s_{11}, s_{12}, s_{13}\}$. The combinations of the previous elements taken k by k are $\{(s_{11}, s_{12}), (s_{11}, s_{13}), (s_{12}, s_{13})\}$. Therefore, the exchanges allowed for each of these combinations are :

$$\begin{array}{lll} s_{11}s_{12} \leftarrow 1 & s_{11}s_{13} \leftarrow 1 & s_{12}s_{13} \leftarrow 1 \\ s_{11}s_{12} \leftarrow 2 & s_{11}s_{13} \leftarrow 2 & s_{12}s_{13} \leftarrow 2 \end{array}$$

The complete $k = 2$ neighborhood for $r = 1, t = 0$, noted $N_2(S_0, 1)$ contains six solutions:

$$(1,1,0,1,0,2) \quad (1,0,1,1,0,2) \quad (0,1,1,1,0,2)$$
$$(2,2,0,1,0,2) \quad (2,0,2,1,0,2) \quad (0,2,2,1,0,2)$$

Notice that the two selected commodities *take the same exchange value*. In fact, the neighborhood is built by packaging two commodities (k products in the general case) currently routed trough the same hub and re-routing them together as a single shipment trough all the alternative paths. However, althougt the sampling procedure reduces the neighborhood size, the exploration time for a given neighborhood still grows very fast with the neighborhood order, therefore limiting in practice the value of k.

4.4 Merging TS and VNS

This section describes the practical implementation of the theorical TS and VNS concepts into a hybrid heuristic in the context of distribution networks as described in section 1. Although TS and VNS are defined herabove as indepent procedures, they are combined into a single algorithm. The search algorithm is the best improvement tabu search heuristic described in section 4.1 operating on a neighborhood chosen by VNS. Therefore, VNS selects a neighborhood structure for the search algorithm based on the status of the search. In practice, this means that VNS is defined by a set of neighborhoods, a criterion to determine when to perform a change of neighborhood, and a rule of selection among them.

We explained in section 4.3 how to derive, from the basic neighborhood, a set of k_{max} different neighborhoods of increasing order. The parameter k_{max} must be adjusted depending of the size of the problem. We present in section 5 several insigths related to the choice of this parameter.

Among the different possible strategies, the heuristic performs a change of neighborhood when a local optimum is reached. During the search process, the best value found so far since the beginning of the exploration of the current neighborhood is recorded in *local_best*. The heuristic considers that a local optimum is reached when no improvement in *local_best* has been achieved after a given number of iterations $it_local[k]$. Notice that the allowed number of iterations without improvement, $it_local[k]$, depends on the type of neighborhood, k. The heuristic keeps trace of both the current, k, and the highest, k_m, value of k used so far. When a change of neighborhood is performed, search continues from the current solution. The objective value of the current solution becomes the *local_best* value for the search on the new neighborhood.

Selection of the new neighborhood after a change depends of the status of the search. Neighborhoods are sorted acording to their order and selected successively by the heuristic. Note that the size of the neighborhood increases with the value of k and so does

the computational effort to explore it. Therefore, it seems an efficient strategy to quit a complex neighborhood and return to the basic one as soon as *local_best* is improved, reseting k to 1. The basic rule for neighborhood selection is that any change on a current neighborhood having order $k > 1$ drives automatically back to the basic structure. However, two cases must be considered when the allowed number of iterations is reached without any improvement on *local_best*. In the first case, the search has been performed on a neighborhood of type k >1 and the heuristic returns back to the basic neighborhood. The order k is stored in k_m as the highest type explored. This strategy assumes that even if the moves performed on the complex neighborhood have not been able to generate any improvement, the search may have escaped from the local optimun where it was trapped. In the second case, when the search has been performed on the order 1 neighborhood, the heuristic jumps to a neighborhood of order $k + 1$. However, if $k + 1$ reaches k_{max} the diversification procedure is called and both k and k_m are set back to 1. Figure 5 lists the pseudocode that implements the rules for neighborhood exchange.
Local search is performed by the TS described in section 4.1. TS keeps track of the best solutions (*local_best* and *global_best*) and their objective values. TS also provides the iteration counters (*it_local* and *it_global*) that help determine the status of the search and the stopping criterion.

5 Computational Results

This section evaluates the performance of the hybrid heuristic TS/VNS by comparing the solutions that it produces to either the optimal solution or the best bound provided by the MIP program given in section 3 when solved with Cplex 6.5. All programs were coded in C++. Tests were run on Ultra-Sparc 10 workstations. All the execution times are given in CPU seconds.

As mentioned in section 4.1, the TS/VNS heuristic incorporates some parameters that must be adjusted prior to the computational test phase. The right choice of values for these parameters can have an important impact on the heuristic performance. However, there is no deterministic rule that provides the best values to those parameters. Therefore, a preliminary phase of experiments was carried out on similar problems (but different instances than the ones described hereabove) to calibrate them. The settings reported in table 2 were the result of this calibration phase in which they were found to provide a good balance between solution quality and computation time. These are the settings used in all the following experiments.

Several pseudo-realistic problems were generated. Each problem proposes a situation where several locations exchange some goods. Locations were extracted from the list of

```
1     Start
2     Generate random initial solution
3     Set k ← 1, k_m ← 1
4     Set it_local ← 1, it_total ← 1
5     do while it_total ≤ it_total_max
6           if it_local ≤ it_local_max[k] then
7                 Local search on neighborhood type k
8                 If improvement is reached then
9                       k ← 1
10                      k_m ← 1
11                      it_local ← 1
12                else
13                      it_local ← it_local + 1
14          else
15                it_local ← 1
16                if k ← 1 then
17                      k ← k_m + 1
18                      k_m ← k
19                      if k ← k_max then
20                            Proceed Diversification
21                            k_m ← 1
22                else
23                      k ← 1
24                k_m ← k
25          it_total ← it_total + 1
26    repeat
27    end
```

Figure 5: Dynamic exchange of the neighborhood order

the most important cities in the U.S.A. ranked by theirs annual transport figures. The network is complete in the sense that direct shipments are allowed between every two locations. Shipment consolidation is also allowed trough a limited number of hubs (i.e. a subset of the locations). In this case, at most one transshipment is allowed between origin and destination.

The testbed is divided into six families of instances depending on network size and how demand was generated. Two types of random demands were used: type A demands were generated by a random uniform distribution, while type B used a random normal distribution weighted by the annual transport figures of each location. Half of the instances of each type were generated with a mean value for the random distribution of σ=300 and the rest with a higher mean value of σ=600. The problems were normalized in order to obtain similar total demands for both sets A and B (for given network size

Neighborhood specific	Order "1" Neighborhood	Order "k" Neighborhood
Tabu List	Number of source nodes	Number of source nodes/5
Iterations without improvement	50	50/k
Origin nodes in sampling	Number of source nodes/5	Number of source nodes/5
Highest k value		3
General parameters		
Stopping criterion	Maximum alloted CPU time	
Diversification step size	Number of commodities/5	

Table 2: TS/VNS parameters

and mean demand).

We considered networks of $10, 20$ and 40 locations, each one acting simultaneously as an origin and as a destination node, sending and receiving goods to and from all other locations in the network. The number of commodities to be routed in a N node network is therefore $N(N-1)$. The maximun number of locations was bounded by the solving capability of CPLEX. Recall that in the context of this paper, solving a distribution network requires the determination of how many hubs and which ones should be opened. However, even for small sized networks, CPLEX could not handle the MIP instances corresponding to this version of the problem. Although the TS/VNS heuristic does not have any problem tackling all these decision levels simultaneously, in order to get at least some lower bounds to assess the quality of its solutions we decided to fix, for each problem, the set H of potential hubs. We limited the number of available hubs to: $h = Card(H) = 2, 3$ and 4, for problems with 10 nodes, $h = 3, 4$ and 5 for those with 20 nodes and $h = 4, 5$ and 6 hubs for the 40 location problems. For each case, several instances were generated by changing the specific locations included in H. The triplet (N, h, H) therefore determines the network structure of the problem. Note that the potential hubs in H were selected arbitrarily from the locations already in the network according to both their geographical situation and their annual transport ranking.

Table 3 summarizes the instances generated for the computational experiments. Problems are grouped by the number of locations in the network, N, the number of available hubs, h, and a specific choice of potential hubs H. The choice of the h hubs in the sets I, II and III depends on the network size, so the set I used with the 10 location problems might be different than the set I associated to the 20 location problems. For each network, a given number of instances (#) were generated using the different combinations of random demand type (A and B) and average demand σ (300 and 600).

Network Stucture			Demand Type							
			A				B			
			$\sigma = 300$		$\sigma = 600$		$\sigma = 300$		$\sigma = 600$	
N	h	H	#	Instances	#	Instances	#	Instances	#	Instances
10	2	I	5	10A2_1 to 5	5	10A2_6 to 10	5	10B2_1 to 5	5	10B2_6 to 10
		II	5	10A2_11 to 15	5	10A2_16 to 20	5	10B2_11 to 15	5	10B2_16 to 20
		III	5	10A2_21 to 25	5	10A2_26 to 30	5	10B2_21 to 25	5	10B2_26 to 30
10	3	I	5	10A3_1 to 5	5	10A3_6 to 10	5	10B3_1 to 5	5	10B3_6 to 10
10	4	I	5	10A4_1 to 5	5	10A4_6 to 10	5	10B4_1 to 5	5	10B4_6 to 10
20	3	I	5	20A3_1 to 5	5	20A3_6 to 10	5	20B3_1 to 5	5	20B3_6 to 10
		II	5	20A3_11 to 15	5	20A3_16 to 20	5	20B3_11 to 15	5	20B3_16 to 20
		III	5	20A3_21 to 25	5	20A3_26 to 30	5	20B3_21 to 25	5	20B3_26 to 30
20	4	I	5	20A4_1 to 5	5	20A4_6 to 10	5	20B4_1 to 5	5	20B4_6 to 10
		II	5	20A4_11 to 15	5	20A4_16 to 20	5	20B4_11 to 15	5	20B4_16 to 20
		III	5	20A4_21 to 25	5	20A4_26 to 30	5	20B4_21 to 25	5	20B4_26 to 30
20	5	I	5	20A5_1 to 5	5	20A5_6 to 10	5	20B5_1 to 5	5	20B5_6 to 10
		II	5	20A5_11 to 15	5	20A5_16 to 20	5	20B5_11 to 15	5	20B5_16 to 20
		III	5	20A5_21 to 25	5	20A5_26 to 30	5	20B5_21 to 25	5	20B5_26 to 30
40	4	I	1	40A4_1	1	40A4_2	1	40B4_1	1	40B4_2
		II	1	40A4_3	1	40A4_4	1	40B4_3	1	40B4_4
		III	1	40A4_5	1	40A4_6	1	40B4_5	1	40B4_6
40	5	I	1	40A5_1	1	40A5_2	1	40B5_1	1	40B5_2
		II	1	40A5_3	1	40A5_4	1	40B5_3	1	40B5_4
		III	1	40A5_5	1	40A5_6	1	40B5_5	2	40B5_6
40	6	I	1	40A6_1	1	40A6_2	1	40B6_1	1	40B6_2
		II	1	40A6_3	1	40A6_4	1	40B6_3	2	40B6_4
		III	1	40A6_5	1	40A6_6	1	40B6_5	2	40B6_6

Table 3: Random generated problems

Computational results of the TS/VNS heuristic over the whole testbed are presented in Table 4. Each line in the table corresponds to aggregate results of instances having the same number of locations and hubs (i.e. aggregation over hub subsets H and average demand levels σ). It also presents the results produced by CPLEX. The data in the table was compiled as follows. For each instance, CPLEX ran until finding an optimal solution or until exhausting a predetermined maximun alloted computation time. We decided arbitrarily to consider as optimal any solution whose gap with respect to the best lower bound was smaller than 0.1%. The maximun computation time was set according to the size of the problems at: 3600, 18000 and 72000 seconds for the 10, 20 and 40-node instances respectively. The first part of the table reports the results obtained by the CPLEX branch–and–bound code. Since for many instances it did not produce integer solution within the alloted time, we report under heading *Succ* the number of times it did succeed in finding at least one. We also report the average values over these successful cases for the first integer solution gap, *F_Gap*, with respect to the best bound obtained at termination, and the time at which the first integer solution was found, *F_Time*. The next three columns present the average, minimum and maximum gaps obtained at termination over all instances in the group (headings *Ave*, *Min* and *Max*). When no values are listed in this part of the table, this means that CPLEX did not provide any integer solution. The second part of the table presents the TS/VNS results. The procedure was run 5 times on each instance, starting with random initial solutions. Under the heading *Gap* we list the average gap of the best solution found by the heuristic with respect to the best bound provided by CPLEX taken over all instances and runs (*Ave*), and the difference between the best and worst gaps observed among the five runs, averaged over all instances in the group (*Diff*). Finally, column *Time* reports the average time at which the best solution was found. The alloted computation times for the TS/VNS heuristic were of course much shorter, they were set to 600, 1800 and 3600 seconds for the 10, 20 and 40-nodes instances respectively.

As can be seen in Table 4, the TS/VNS algorithm performs extremely well, improving in all the cases over the average integer solutions produced by CPLEX (when available). Gaps were systematically lower than 6% (with respect to the linear relaxation) for the 10 and 20 location problems. Colunm *Diff* provides a mesure of the heuristic robustness. Here again, the heuristic behaviour is very good for the 10 and 20-location problems with average differences between runs staying well below the 1.5% range. However, both the gaps and *Diff* values obtained for the 40-location problems do not show the same good behaviour with, for exemple, average gap values rising to almost 13% in one case. As we mentionned before, solving this problem exactly is a very difficult task. Over the whole testbed, none of the instances could be solved to optimality by CPLEX. In addition, as size increased, producing integer solutions also proved difficult. In fact, the exact procedure could produce such solutions in only 25 out of the 180 instances with

Problem				Branch & Bound						TS/VNS		
				Integer Sol			Gap			Gap		Time
D	N	H	Inst.	F_Gap	F_Time	Succ	Ave.	Min	Max	Ave	Diff	
		2	30	16.86%	109	30	1.4%	0%	4.67%	1.03%	0.3%	148
A	10	3	10	24.64%	325	10	20.16%	4.53%	27.42%	5.71%	0.23%	152
		4	10	37.12%	535	10	30.46%	16.28%	41.33%	5.68%	0.35%	121
		3	30	22.36%	6158	7	20.56%	17.46%	26.9%	1.7%	0.37%	448
A	20	4	30							3.53%	1.32%	522
		5	30							2.03%	0.68%	475
		4	3							12.75%	2.7%	1352
A	40	5	3							12.02%	4.61%	2048
		6	3							11.75%	6.86%	2044
		2	30	18.72%	79	30	2.28%	0%	12.56%	0.95%	0.11%	122
B	10	3	10	38.97%	356	10	14.03%	5.32%	31.28%	3.84%	0.12%	148
		4	10	27.15%	665	10	23.64%	15.16%	41.85%	5.02%	0.15%	132
		3	30	31.33%	4516	13	29.46%	3.04%	43.39%	4.45%	0.76%	549
B	20	4	30	33.17%	7130	5	33.17%	14.8%	37.77%	4.12%	1.03%	327
		5	30							5.41%	0.84%	318
		4	3							11.29%	10.14%	2727
B	40	5	3							9.52%	7.8%	2231
		6	3							8.75%	8.45%	2053

Table 4: Results for random generated problems

20 locations and for none of the 40 locations. Furthermore, some instances are clearly more difficult to solve than others as can be observed by comparing the *Min* and *Max* columns. The different between these two values (when available) goes from 4.67% up to 40.35% for the 10B4 family.

In order to assess if the larger gaps were due to the heuristic producing lower quality solutions or to a decrease in quality of the lower bounds provided by CPLEX, we decided to carry out additional experiments on a subset of the larger problems in which these bigger gaps and *Diff* values were observed. For these experiments we selected 9 instances of 40 nodes and we raised the time limit imposed on the CPLEX runs to 250000 seconds. We also extended the running time for the TS/VNS up to 10800 seconds (3 times the original alloted time). In Table 5, we report the results of these "extended-time" experiments and compare them to the previous. The first part of the table shows the results for the exact method mesured in terms of gap variation with respect to the best bound obtained from the previous runs (with a maximum CPU time of 72000 seconds). The first column gives the reduction in gap for the root node to the 72000 seconds mark while the second lists the further gap improvement obtained between the 72000 seconds and the 250000 seconds marks. The second part of the table gives, for each instance considered, the average best and worst gaps observed over the 5 runs again with respect to the 72000 seconds CPLEX lower bound. Finally, we took the worst run of the TS/VNS heuristic for each instance as observed in the previous experiments (with a time limit of 3600 seconds) and extended its time limit three and fivefold. The two last columns of the table report the gaps for the solution thus obtained under headings 10800 and 18000 (seconds).

	B&B Bounds Imp		Gaps TS/VNS			Worst case extended runs	
Problem	Root node	250000	Ave	Best	Worst	10800	18000
40B4_1	6.74%	0.03%	13.75%	9.85%	23.04%	3.88%	3.75%
40B4_2	7.12%	0.03%	16.81%	9.92%	23.36%	11.71%	3.82%
40B4_3	6.16%	0.03%	3.34%	1.9%	5.71%	3.01%	3.01%
40B5_1	8.75%	0.06%	12.94%	5.34%	20.68%	5.59%	5.71%
41B5_2	9.25%	0.05%	4.28%	3.76%	4.92%	4.92%	4.92%
42B5_3	8.74%	0.07%	11.21%	8.48%	14.5%	2.63%	2.63%
40B6_1	9.64%	0.01%	6.41%	5.89%	7.19%	5.96%	5.96%
41B6_2	10.27%	0%	2.59%	1.88%	3.3%	3.3%	3.3%
42B6_3	9.48%	0.27%	17.27%	6.07%	28.72%	5.93%	5.93%

Table 5: Results of longer tests for the 40-location subset

As can be seen, even with the extra-time, CPLEX did not find any integer solution. Moreover, in all the cases the improvement of the lower bound was extremely small. The reason of this poor performance could be that we kept the default setting for the MIP solver of CPLEX during all the tests. This default setting does not seem to be adequate for solving the proposed set of problems. However, the development of efficient branching rules or even the improvement of the MIP model by the addition of new constraints are issues beyond the scope of this paper which could warrant several research projects in themselves. On the other hand, the TS/VNS heuristic was able to improve its best values in 7 out of the 9 instances during the first time extension. In fact, a more detailled analysis shows that the TS/VNS heuristic did improve the quality of all the "low quality" solutions. However for the "good solutions" (i.e. those with the gap of less than 5%) found witin the regular time runs could not be improved. Note that over the complete time extension, all gaps did drop under 6%. Such small differences attest to the quality of both the heuristic solutions and the lower bounds produced by CPLEX, as well as to the extreme difficulty experienced by standard exact procedures in producing integer solutions.

Though the performance of the TS/VNS heuristic is clearly good, the efficiency of the algorithm should also be compared to other existing heuristic approaches. Recall that the development of the TS/VNS heuristic was motived by the hope of improving on both TS and VNS when considered individually. The natural choice is then to compare the hybrid algorithm with respect to its two main components with the same running times and initial conditions. We performed 5 runs of each metaheuristic on a representative subset of the main testbed containing 20 and 40 location instances. Table 6 reports the results observed on this subset of instances mesured in terms of the gap between the average solutions found by each of the three methods.

Table 6 shows that TS produced better solutions than VNS in all the cases. However both heuristics were dominated in all the cases by the hybrid TS/VNS. Even if the difference between the results produced by the TS and the TS/VNS heuristics are close to zero in several cases, they assess to the efficiency of the hybrid approach.

6 Conclusions

In this paper, a model for designing distribution networks with transhipment centers was presented. The model can handle several transportation modes and considers explicitly two commodity attributes (weight and volume). It was shown that in such a framework, the optimal choice between modes depends of the amount of flow to transport and the

Instance	Time	Average Gap		
		TS/VNS	VNS	TS
20A3_1	600	0%	0.86%	0.01%
20A3_6	600	0%	3.71%	2.3%
20A4_1	600	0%	2.88%	0.7%
20A4_6	600	0%	1.37%	0.2%
20A5_1	600	0%	1.66%	1.27%
20A5_6	600	0%	0.95%	0.04%
40A4_1	3600	0%	3.88%	0.84%
40B4_1	3600	0%	4.56%	4.47%
40A5_1	3600	0%	4.09%	0.26%
40B5_1	3600	0%	4.76%	0.16%
40A6_1	3600	0%	4.65%	0.34%
40B6_1	3600	0%	4.18%	3.46%

Table 6: TS/VNS versus TS and VNS

resulting "optimal" cost function can be modeled by a piecewise linear function. Such functions can be easily integrated within a mixed integer program. However, the size of the associated program increases very fast with the size of the problem to solve, overcoming the capacity of solvers even for small to medium sized instances. A heuristic approach was therefore proposed: tabu search and variable neighborhood search, two metaheuristics that have proven their effectiveness in solving large combinatorial problems, were merged into a new hybrid metaheuristic. The solutions produced by the TS/VNS heuristic were compared to those produced by the exact method on a testbed generated with real data (nodes and tariffs) and random demands. The results attest to the excellent quality of the heuristic solutions and comfirm the extreme difficulty of the problem. Finally, the efficiency of the TS/VNS was also tested against both TS and VNS considered individually and was found superior in all the cases tested.

References

[1] T. Aykin. Lagrangean relaxation approaches to the capacited hub and spoke network design problem. *European Journal of Operational Research* **79**, 501–523, 1994.

[2] C. Barnhart, E.L. Johnson, G.L. Nemhauser, G. Sigismondi, P. Vance. Formulating a mixed integer programming problem to improve solvability. *Operations Research*

41, 1013–1019, 1993.

[3] L.G. Barton and P. Rech. A non-convex transportation algorithm. In E.M.L. Beale, editor, *Applications of Mathematical Programming Techniques*, pages 250–260. American Elsevier Publishing Company, INC, 1970.

[4] L.D. Burns, R.W. Hall, D.E. Blumfeld, and C.F. Daganzo. Distribution strategies that minimize transportation and inventory costs. *Operations Research* **33**, 469–490, 1985.

[5] J.D. Camm, T.E. Chorman, F.A. Dill, J.R. Evans, D.J. Sweeney, and G.W. Wegryn. Blending OR/MS, judgment, and GIS: restructuring P&G's supply chain. *Interfaces* **27:1**, 128–142, 1997.

[6] J.F. Campbell. Integer programming formulations of discrete hub locations problems. *European Journal of Operational Research* **72**, 387–405, 1994.

[7] J. Ebery, M. Krishnamoorthy, A.T. Ernst, and N. Boland. The capacited multiple allocation hub location problem : Formulations and algorithms. *European Journal of Operational Research* **120**, 614–632, 2000.

[8] A.T. Ernst, and M. Krishnamoorthy. Efficient algorithms for the uncapacited single allocationp-hub median problems. *Location Science* **4**, 139–154, 1996.

[9] A.T. Ernst, and M. Krishnamoorthy. Exact and heuristic algorithms for the uncapacited multiple allocation p-hub median problem. *European Journal of Operational Research* **104**, 100–112, 1998.

[10] B. Fleischmann. Designing distribution systems with transport economies of scale. *European Journal of Operational Research* **70**, 31–42, 1993.

[11] A.M. Geoffrion and G.W. Graves. Multicommodity distribution system design by Benders decomposition. *Management Sciences* **20**, 822–844, 1974.

[12] F. Glover. Tabu search–part I. *ORSA Journal on Computing* **1**, 190–206, 1989.

[13] F. Glover. Tabu thresholding: Improved search by nonmonotonic trajectories. *ORSA Journal on Computing* **7**, 426–442, 1995.

[14] R.W. Hall. Dependence between shipment size and mode in freight transportation. *Transportation Science* **19**, 436–444, 1985.

[15] R.W. Hall. Route choice on freight networks with concave cost and exclusive arcs. *Transportation Research B* **23**, 177–194, 1989.

[16] R.W. Hall. Route selection on freight networks with weight and volume constraints. *Transportation Research B* **25**, 175–189, 1991.

[17] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *Les cahiers du GERAD* **G–98–20**, Montral, Canada, 1998.

[18] P. Hansen and N. Mladenović. Recherche voisinage variable. *Les cahiers du GERAD* **G–2000–08**, Montral, Canada, 2000.

[19] D.L. Kelly and B.M. Khumawala. Capacited warehouse location with concave costs. *Journal of the Operational Research Society* **33**, 817–826, 1982.

[20] J.G. Klincewicz. Solving a freight transport problem using facility location techniques. *Operations Research* **38**, 99–109, 1990.

[21] J.G. Klincewicz. Heuristics for the p-hub location problem. *European Journal of Operational Research* **53**, 25–37, 1991.

[22] J.G. Klincewicz. Avoiding local optima in the p-hub location problen using tabu search and grasp. *Annals of Operations Research* **40**, 283–302, 1992.

[23] B.W. Lamar. An improved branch and bound algorithm for minimum concave cost network flow problems. *Journal of Global Optimization* **3**, 261–287, 1993.

[24] R.F. Love, J.G. Morris,and G.O. Wesolowsky. *Facilities location: Models and methods*. North Holland, Amsterdam. 1988.

[25] M. O'Kelly. Hub facility location with fixed cost. *Papers in Regional Science* **71**, 293–306, 1992.

[26] M. O'Kelly, D.L. Bryan, D. Skorin-Kapov and J. Skorin-Kapov. Hub network design with single and multiple allocation: a computational study. *Location Science* **4**, 125–138, 1996.

[27] M. O'Kelly, and D.L. Bryan. Hub location with flow economies of scale. *Transportation Research B* **32**, 605–616, 1998.

[28] H. Pirkul and D.A. Schilling. An efficient procedure for designing single allocation hub and spoke systems. *Management Science* **44**, 235–242, 1998.

[29] W.B. Powell and Y. Sheffi. Design and implementation of an interactive optimization system for network design in the motor carrier industry. *Operations Research* **37**, 12–29, 1989.

[30] D. Skorin-Kapov, and J. Skorin-Kapov. On tabu search for the location of interacting hub facilities. *European Journal of Operational Research* **73**, 502–509, 1994.

[31] D. Skorin-Kapov, J. Skorin-Kapov, and M. O'Kelly. Tight linear programming relaxations of uncapacited p-hub median problems. *European Journal of Operational Research* **94**, 582–593, 1997.

[32] L. Vogt and J. Even. Piecewise linear programming solutions of transportation costs as obtained from rate tariffs. *American Institute of Industrial Engineering Transactions* **4**, 148–153, 1972.

[33] J.B. Yaged. Minimun cost routing for static network models. *Networks* **1**,139–172, 1972.

[34] W.I. Zangwill. Minimum concave cost flows in certain networks. *Management Science* **14**, 429–450, 1968.